2X−8

文字の出現頻度を考慮した文字列検索アルゴリズムの提案

大曽根　匡、佐藤　創
(専修大学経営学部情報管理学科)

1．はじめに

近年、文献情報等の文書情報のＤＢ化が急速に進められているのに伴い、文書情報処理の高速化のニーズが極めて高まっている。このような状況の中で、文書情報処理のうち最も基本的かつ高負荷な処理の一つであるストリング・サーチ処理の高速化は重要な課題である。その高速化を目的としたアルゴリズムとして、ＫＭＰ法やＡＣ法、ＢＭ法[1]などが著名である。また、ＡＣ法とＢＭ法をハイブリッドしたアルゴリズム[2][3][4]も提案されている。しかし、これらのアルゴリズムはテキスト上の文字の出現頻度について余り考慮されていなかった。そこで、本稿では、以前提案したアルゴリズム[2]を拡張し、その拡張したアルゴリズムを文字の出現頻度により使い分けるという方法を提案する。また、性能実験により、その有効性についても検証する。

2．アルゴリズム A-N

提案するアルゴリズムの基本的な考え方は、以下の通りである。高速なアルゴリズムとして著名なＢＭ法は、パターンの末尾の文字から照合することにより、高速化を図っている。例えば、パターンが「ＡＢＣＤＥ」の場合は、「Ｅ」から照合を始める。しかし、テキスト上で「Ｅ」の出現頻度が高い場合は、この方法は効率が良くない。むしろ、「Ｄ」から照合を始めたほうが、効率がよいと考えられる。そこで、一般にパターンのＮ文字目から照合を行うアルゴリズムを考え、それをアルゴリズム A-N と呼ぶことにする。照合の順序はいろいろ考えられるが、本アルゴリズムでは、照合が成功している間は、順次１文字ずつ前方に照合を進めていくことにする。パターンの先頭まで照合に成功したら、次は、パターンの末尾文字から前方に照合を進めることにする。途中で照合に失敗したら、ＢＭ法の原理と同様に、できるだけパターンを後方にシフトさせ、再びパターンのＮ文字目から照合を開始する。

提案アルゴリズムでは、[2][3]のアルゴリズムと同様に、これを状態遷移テーブルとスキップテーブルだけを用いて行う。ここで、状態 i は、i 文字分だけ照合に成功している状態とする。状態遷移テーブル T[i,c] は、現在の状態 i と入力文字 c とから、次の状態を導出するためのテーブルである。一方、スキップテーブル S[i,c] は、現在の状態 i と入力文字 c とから、テキスト上で次に何文字先の文字を入力させればよいかというスキップ幅を導出するためのテーブルである。これらは、パターンの情報だけから作成することができる。例として、表１～３に、パターンが「ＡＢＣＤＥ」の場合の、アルゴリズム A-4 に対する状態の定義と状態遷移テーブルとスキップテーブルを示す。

テキスト上で「Ｅ」の出現する頻度が 0.6 の場合の動作例を図１と図２に示す。図１が A-5 を用いたときの動作例で、図２が A-4 を用いたときの動作例である。この例の場合、A-5 はＢＭ法と同じ振る舞いをする。そして、25 文字のテキストに対し、A-5 では 10 文字、A-4 では 8 文字の入力で検索を終了する。すなわち、平均スキップ幅は、A-5 では 2.5文字、A-4 では 3.1 文字となり、A-4 のほうが高速であることがわかる。

3．性能実験

アルゴリズムの性能は、平均スキップ幅によって表現できる。そこで、アルゴリズム A-N (N=1～5) に対し、この平均スキップ幅を見積もるための実験を行った。実験では、パターンを「ＡＢＣＤＥ」とし、特定の文字「Ｅ」と「Ｄ」の出現確率をそれぞれ変化させることにより、各アルゴリズムの平均スキップ幅を比較することにした。ここで、テキスト長は 10000 文字、アルファベットの文字種は 32 文字とし、特定文字以外の文字の出現確率は一様とした。この試行を 100回行い、その平均を求めた。図３と図４に、「Ｅ」と「Ｄ」の出現確率を変化させた場合の結果をそれぞれ示す。

図３より、A-5 は、「Ｅ」の出現確率が増加するにつれ性能が悪くなるが、A-4 や A-3 では、逆に良くなることがわかる。そして、「Ｅ」の出現確率が 0.25 付近以下の場合は A-5 が最も性能が良く、0.25 付近以上の場合は A-4 が最も性能が良いことがわかる。したがって、そこを境界として A-5 と A-4 を使い分けると良いことになる。一方、図２から、「Ｄ」の出現確率の増加が、A-5 や A-4 に対し、悪い影響を与えていることがわかる。そして、「Ｄ」の出現確率が 0.5 付近以下では A-5 が、以上では A-3 が最も性能が良いことがわかる。

String Searching Algorithms with considering the generic distribution of characters.
Tadashi OHSONE and Hajime SATO,
Department of Information Management, Senshu University.

4．まとめ

テキスト上において、パターンの末尾の文字やそれより１つ前の文字の出現確率が高い場合などでは、ＢＭ法のようにパターンの末尾の文字から照合して行く方法が効率的でないことがわかった。そこで、パターンのＮ文字目から照合を始めるアルゴリズム A-N を提案した。そして、常にＢＭ法やそれに類似した方法を用いるのではなく、文字の出現確率に応じて、今回提案したアルゴリズム A-N を使い分ける方法が効果的であることを性能実験により検証した。

表１．状態の定義 (A-4)

状態番号	状　　態
0	△△△＊△△△△△△
1	△△＊D△△△△△△
2	△＊CD△△△△△△
3	＊BCD△△△△△△
4	ABCD＊△△△△△
5	ABCDE△△△＊△

＊：入力位置

表２．状態遷移テーブル (A-4)

	A	B	C	D	E	#
0				1		
1			2			
2		3				
3	4					
4					5	
5				1		

空白：0　　#：その他の文字

表３．スキップテーブル (A-4)

	A	B	C	D	E	#
0	3	2	1	-1	4	4
1	5	5	-1	5	5	5
2	6	-1	6	6	6	6
3	4	7	7	7	7	7
4	3	4	4	4	4	4
5	3	2	1	-1	4	4

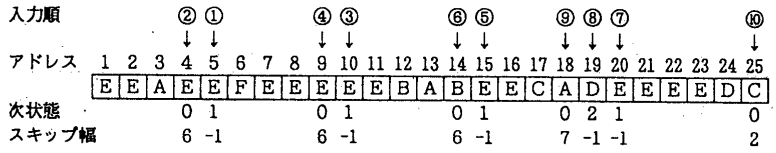

図１．アルゴリズム A-5 の動作例

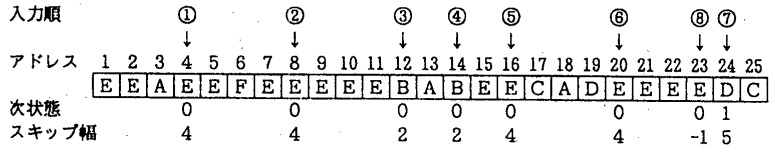

図２．アルゴリズム A-4 の動作例

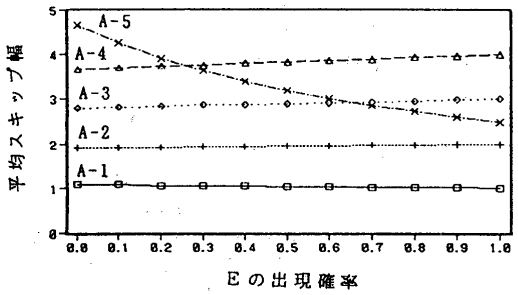

図３．Ｅの出現確率に対する平均スキップ幅

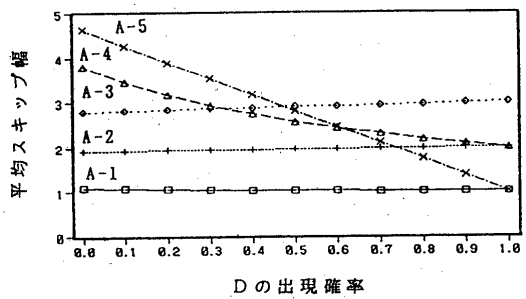

図４．Ｄの出現確率に対する平均スキップ幅

参考文献

【1】R.S.Boyer and J.S.Moore,"A Fast String Searching Algorithm," Comm.ACM, Vol.20, No.10, pp.762-772 (1977).

【2】大曽根 他,"高速ストリングサーチアルゴリズムの提案," 情報処理学会第34回全国大会, pp.463-464 (1987).

【3】大曽根 他,"複数パターンに対する高速ストリングサーチアルゴリズムの提案," 情報処理学会第35回全国大会, pp.49-50 (1988).

【4】浦谷,"複数文字列照合アルゴリズム," 情報処理学会第35回全国大会, pp57-58 (1988).